IN THE CLAIMS

1. (Currently Amended) A system that allows analysis of software running in a tamper-resistant environment, the system comprising:

a processor which monitors at least one instance of software execution ~~activities~~ identified and selected by a user to be monitored and creates a log entry with ~~for~~ at least one set of data derived from the one instance of software execution, whereby the set of data is used to diagnosis the software execution;

an encryption system which encrypts the log entry for the at least one set of data ~~selected activity~~;

a log file of a relatively-fixed size which stores the ~~encrypted~~ log entry for the at least one set of data ~~entries~~ which have been encrypted; ~~and~~

random data in the log file when it is originally created and which is replaced by log entries so that ~~the~~ a size of the log file ~~containing~~ including log entries appears to be a substantially-constant size; and

a pointer which ~~identified~~ identifies ~~the~~ a next storage location for ~~the~~ a next log entry so that ~~the~~ a last log entry can be determined and the next log entry can be positioned in a location in the log file after ~~the~~ a previous log entry.

2. A system including the elements of Claim 1 wherein the system includes a transmission system for sending the log file, upon command, to a secure processing location away from the system in which the log file was created.

3. (Currently Amended) A system including the elements of Claim 1 wherein the system includes a system for wrapping around and filling the log file from ~~the~~ a beginning when the log file has been filled, allowing the log file to remain at a substantially-constant size even after the log file has been filled with data and a new entry is received.

4. (Currently Amended) A system including the elements of Claim 1 wherein the system includes a mechanism for obscuring the ~~activity for which a~~ log entry ~~is~~ which has been created.

5. (Currently Amended) A system including the elements of Claim 4 wherein the mechanism for obscuring the ~~activity for which a~~ log entry ~~is~~ which has been created includes a printing function for writing into the log file.

6. (Currently Amended) A system including the elements of Claim 2 wherein the system includes a mechanism for receiving an indication from ~~the~~ a user that transmission is desired and transmits the log file in response to that indication.

7. (Currently Amended) A system including the elements of Claim 1 wherein the system further includes a mechanism for receiving an input from a user that initiates logging of log entries into the log file each time logging is desired by the user.

8. (Currently Amended) A system including the elements of Claim 1 wherein the system further includes an initializing mechanism for determining ~~when~~ each instance logging is to begin and initiating logging of log entries only in response to that initializing mechanism.

9. (Currently Amended) A system including the elements of Claim 1 wherein the system uses a public key to ~~provide~~ encrypt the log ~~entries~~ entry which has been created and a private key corresponding to the public key is used to decrypt the log ~~entries~~ entry which has been created at a secure location.

10. (Currently Amended) A method ~~of analyzing the operation of~~ for diagnosing software in a tamper-resistant environment comprising the steps of:

generating a log file full of random data;

turning on logging and establishing a pointer for ~~the~~ a location of ~~the~~ a next logged ~~event~~ software operation activity;

monitoring ~~the~~ at least one ~~operation of~~ software operation activity within the tamper-resistant environment and generating messages in response to ~~operation of the~~ at least one instance of software execution within the tamper-resistant environment;

logging ~~an event~~ at least one software operation activity relating to a generated message by replacing a random data with an encrypted record of ~~an event~~ the software operation activity;

moving the pointer when a log entry has been made to ~~the~~ a next available log position;

wrapping the pointer to ~~the top~~ a beginning of the log file when the log file is full of log entries; and

sending the log file to a secure location where ~~it may~~ the log file can be decrypted and analyzed; and

analyzing decrypted log file data and providing information ~~on the operation of the~~ for diagnosing software in the tamper-resistant environment.

11. (Currently Amended) A method including the steps of Claim 10 wherein the step of turning on logging includes the steps of receiving ~~an~~ a user input indicating that logging is desired and initiating the logging in response thereto.

12. (Currently Amended) A method including the steps of Claim 10 wherein the step of ~~logging an event~~ at least one software operation activity further includes the steps of determining whether the ~~event~~ software operation activity is to be logged, and if so, determining when to ~~log~~ encrypt the ~~event~~ software operation activity to obscure what is being logged.

13. (Currently Amended) A method including the steps of Claim 10 wherein the step of logging ~~an even~~ the software operation activity further includes the steps of determining ~~the next location for logging~~ a next available log position, replacing ~~the~~ existing data in the ~~location~~ next available log position with ~~the~~ data from the ~~event~~ software operation activity, and updating the pointer to provide ~~the~~ a location of ~~the~~ next logged ~~event~~ software operation activity.

14. A method including the steps of Claim 10 and further including the step of receiving a command from a user that indicates that sending the log file to a remote location is desired and transmitting the log file in response thereto.

15. (Currently Amended) A ~~service which operates to analyze~~ method of analyzing the operation of software in a remote protected processing environment, the ~~service~~ method including:

receiving from the remote protected processing environment an encrypted log file of substantially-constant size ~~representing~~ comprising at least one log ~~entries~~ entry with ~~of selected events~~ at least one set of data derived from at least one instance of software execution monitored in response to a user identifying and selecting the one instance of software execution, whereby the set of data is used to diagnose the software execution ~~which occurred at the remote protected processing environment~~;

determining a decrypting key for the encrypted log file and decrypting the encrypted log file;

analyzing the log ~~entries~~ entry ~~of selected events~~ at the remote protected processing environment ~~and~~ to ~~determining~~ determine whether ~~the~~ an operation of the remote protected processing environment corresponding to the at least one set of data derived from at least one instance of software execution is appropriate; and

reporting ~~the~~ results of the analyzing step.

16. (Currently Amended) A ~~service~~ method providing the steps of Claim 15 and further including providing an instruction to initiate ~~the~~ a logging of messages each time logging is desired by the user and an instruction to send to the encrypted log file to ~~the~~ a remote ~~location~~ system for analysis.

17. (Currently Amended) A ~~service~~ method providing the steps of Claim 16 wherein the instruction to initiate logging of messages includes the step of initiating programming within the ~~remote system~~ remote protected processing environment to replace information in ~~a~~ the encrypted log file with encrypted information relating to the operation of the remote protected processing environment.

18. (Currently Amended) A ~~service~~ method providing the steps of Claim 17 wherein the step of replacing ~~data~~ information in the encrypted log file includes the step of replacing random data which was placed in the encrypted log file when it was created.

19. (Currently Amended) A ~~service~~ method providing the steps of Claim 17 wherein the step of replacing ~~data~~ information in the encrypted log file includes the step of using a pointer to ~~the~~ a next location in the encrypted log file and the pointer wraps to ~~the top~~ a beginning of the log file after the encrypted log file has been filled.

20. (Currently Amended) ~~Software stored on a device comprising~~ A computer program product for analyzing software running in a tamper-resistant environment, the computer program product comprising instructions for:

~~a first module including stored program instructions for~~ recording ~~events~~ at least one set of data serviced from at least one instance of software execution identified and selected by a user to be monitored whereby the set of data is used to diagnosis the software execution;

~~a second module for~~ encrypting the recording of ~~events~~ the at least one set of data using a key;

~~a third module for~~ recording the at least one set of data, which has been encrypted ~~events~~ sequentially in a storage block of a substantially fixed size;

~~a fourth module~~ maintaining a pointer ~~of the~~ to a next available location for ~~the log~~ recording the at least one set of data sequentially in the storage block;

~~a fifth module for~~ responding to a command and sending ~~the~~ an encrypted log file comprising the at least one set of data which has been encrypted and sequentially recoded in the storage block to a remote location for decryption and analysis.

21. (Currently Amended) ~~Software including the elements of Claim 20 wherein the software further includes~~ The computer program product of claim 20, further comprising instructions for:

~~a mechanism for~~ initializing the storage block of a substantially fixed size with random information which has been encrypted to provide a block of apparent data.

22. (Currently Amended) ~~Software including the elements of Claim 20 wherein the software further includes~~ The computer program product of claim 20, further comprising instructions for: ~~a module for~~ writing the at least one set of data which has been encrypted and recorded ~~events~~ in a sequential order in the ~~fixed-size~~ storage block of the substantially fixed size and for wrapping around when ~~the~~ an end of the ~~fixed-size memory~~ storage block of the substantially fixed size is reached.